

IN THE SPECIFICATION:

Please revise the specification as follows:

Please amend paragraph numbers 0054, 0062, 0065, 0066, 0072, 0076, 0077, 0080, 0098, 00101, 00102, 00116, 00143, and 00152 as follows:

[0054] In some embodiments, the conversation management system 102 creates a snippet of the conversation at step 320. The snippet provides the user with a preview of the contents of the conversation without the user having open the conversation. In one embodiment, the snippet is generated from the most recent message in the conversation. In another embodiment, the snippet is extracted from the first message (i.e., the oldest message) in the conversation. In yet another embodiment, the snippet is extracted from all the messages in the conversation according to predefined heuristic rules, e.g., listing a set of keywords appearing most frequently in the conversation. If the conversation management system 102 is preparing a list of conversations in response to a search submitted by the user, it creates a snippet including a highlighted portion that matches the user-submitted query terms at step 322, which may be similar in one or more respects to the snippets included search results returned by a search engine such as the GoogleTM (trademark of Google Inc.) search engine.

[0062] Fig. 5A is a flowchart illustrating the operation of the “add labels” pull-down list 345. After a user clicks on the pull-down list, the browser displays a list of user-defined labels for him to choose at step 502. The user selects a particular label in the pull-down list and at least one conversation in the conversation list at step 504. The browser then associates the user-selected conversations with the user-selected label at step 506 and sends them back to the conversation management system for further processing at step 508. As a result, a list of the conversations associated with the user-selected label will include the conversations associated with that label in step ~~[[506]]~~ 510. As described in more detail below, when a label or attribute is associated with a conversation, one or more entries are added to an index 1710 (Figs. 17, 18) to enable searches for conversations bearing that label to be located. In some embodiments, the label or attribute is also added to one or more entries in a conversation database 1808 (Fig. 18). In some embodiments, the labels associated with a conversation are displayed when the user views the conversation. In some embodiments, the labels associated with each conversation in a list of conversations are displayed in the row of information provided for each conversation in the list.

[0065] The conversation assistant first receives a series of messages belonging to a particular conversation as well as an associated set of display instructions at step 602. The conversation assistant, at step 604, identifies the topic of the conversation as well as any system-defined categories and user-defined labels associated with the conversation. Based on the categories and labels, the conversation, at step 606, generates a list of user-selectable actions and a list of user-selectable labels, respectively. These two lists are similar to the two pull-down lists 344 and 345 shown in Fig. 3B. In one embodiment, the list of user-selectable actions includes only actions that are applicable to the particular conversation (i.e., the conversation to be displayed). Therefore, a conversation that is already in the “Trash” category cannot be associated with the “Trash” category again. By the same token, the list of user-selectable labels only includes labels that have not yet been associated with the particular conversation.

[0066] Starting from step 608, the conversation assistant generates a sub-form for each individual message in the conversation, the sub-form including at least a detailed message header and a message body. Each detailed message header includes the names and/or addresses of the sender and all the primary and second recipients, a set of message-level actions and a date/time value. The conversation assistant also creates a concise message header from the detailed message header at step 610. The concise message includes the identifiers of the sender and the recipients. If the user himself is in the recipient list, the conversation assistant utilizes a unique character string or icon as his identifier at step 610, which is similar to the user’s identifier in the sender list. For example, the pronoun “me” in English can be used to represent the user in the concise message header at step 612. If the message is the first one to identify a particular recipient (i.e., in the primary or secondary recipient fields of the message header) in this conversation, the conversation assistant will highlight the recipient’s identifier in the concise message header at step 614.

[0072] Fig. 6D is a schematic screenshot of the conversation shown in Fig. 6B after the user clicks on the “Expand All Messages” button 634 (or button 743, Fig. 7C). As a result, all ~~[[the]]~~ four messages ~~[[608]]~~ 680 are displayed in the expanded mode. Note that the text in the button 634 changes to “Compact All Messages”. The display of the

conversation returns to the format shown in Fig. 6B after the user clicks on the button 634 again.

[0076] Fig. 7B is a schematic screenshot of a browser window listing three conversations 730 that match the query term “Tahoe” 720. Above the three conversations 730 is a “Search Options” link 732. The user can click on the link to update the options that were used to conduct the search. For example, the user can specify whether the search is limited to conversations in one category or to conversations in all the categories. When the listed conversations belong to two or more distinct categories, none of the categories in group box 346 are highlighted. Within each matching conversation, a snippet that includes the term “Tahoe” 734, 736, 738 is identified and displayed with the word “Tahoe” being highlighted in each of the three conversations 730. These snippets indicate that the three conversations cover three distinct topics, i.e., the first conversation is about a ski trip to Lake Tahoe, the second conversation is about a Tahoe-model SUV, while the third conversation is about camping sites at the Tahoe National Forest. Therefore, the user can use the information in the snippet to decide which particular conversation he wants to open.

[0077] Fig. 7C is a schematic screenshot displaying all [[the]] messages 750 in the third conversation, labeled “Topic M” 740, after the user clicks on it. Note that [[the]] an “All Mail” category 741 has been highlighted (in group box 346) in this screenshot, indicating that the third conversation “Topic M” 740 is currently located in the “All Mail” category 741. Since the second message includes [[the]] a query term “Tahoe” 752, the second sub-form is displayed in the expanded mode with the word “Tahoe” 752 highlighted.

[0080] Fig. 9B depicts a screenshot comprising three separate browsing windows at different stages of the spell check process. The first window 820 contains a spell-checked copy of the message. In particular, three potentially misspelled words 821, 823 and 825 have been highlighted in the window. After the user clicks on the second word 823, the word “delying” is then incorporated into a spell correction window 832 as shown in the second browsing window 830. Further, a list of candidate words 834 generated by the system are displayed under or adjacent the spell correction window. The user then selects the third replacement term, “delaying”, to replace the misspelled term “delying”. If the user wants to add the word “temporarily” before the corrected word “delaying”, the user can click on the left arrow 841 (or right arrow 843) as shown in the third window 840, position

the browser window pointer to a position in front of the misspelled term “delying,” and then enter the word “temporarily.” Finally, after correcting all the misspelled words, the user clicks on the “Done” button 845, which causes the message to be sent to S4.

[0098] Referring to Fig. 12, the following is a more detailed description of how a message may be matched with a conversation (stage [[1106]] 1208). After the message is parsed (stage 1104 of Fig. 11), portions of the constituent parts may be used to formulate a search query for candidate conversations (stage 1202). For example, conversations may be related based on temporal information, sender information, subject information, other information in the header, or system supplied information, or any combination thereof. One embodiment formulates the search query using the subject of the message along with the “References:” and/or the “In-Reply-To:” field. Other embodiments include the sender of the message or the recipients in the search query. One of ordinary skill in the art would recognize many different search queries to produce messages which might be related to the current message.

[00101] Other factors may be considered when matching a message to a conversation such as the recipients, portions of the message content, domain addresses of the sender, or other factors. Those of ordinary skill in the art will recognize that there are many different ways to determine the conversation to which a new message belongs. Once the current message is matched up with a satisfactorily matching message, the conversation associated with the matching message may be associated with the current message (stage [[1206]] 1208). On the other hand, if no satisfactory matching message (and, therefore, no conversation) is found, a new conversation identifier is associated with the message (stage [[1206]] 1208).

[00102] As mentioned earlier, after the current message is associated with a conversation (previously existing or newly created) (stage [[1106]] 1208), initial attributes are added (stage [[1108]] 1112). One embodiment for adding these initial attributes (stage [[1108]] 1112) may be better understood by referring to Fig. 13. Initially, the sender of the message may be examined. If the sender of the current message is not the user for whom this message processing is being performed, it highly likely that the current message has been sent to that user by someone else. Accordingly, if at stage 1302 it is determined that the sender of the current message is not the user, an initial attribute of “InBox” (also called

the InBox label) may be associated with the current message (stage 1304). An attribute of “InBox” may be used to identify those messages which have been sent to the user. For example, in some embodiments a search of messages having the “InBox” attribute is used in the production of an InBox view for the user.

[00116] Once the list of conversations has been created (stage 1510) by identifying the conversations to which all or a portion of the returned results belong, the conversation information is formatted (stage 1512) for return to the requestor (stage [[1512]] 1514). The formatting stage 1512 may format a number of different types of information associated with the identified conversations, the details of which will vary from one embodiment to another. In some embodiments, the formatting information is of the type that is recognized by the Conversation Assistant 110 for presentation of the conversation information to the user.

[00143] Referring to Fig. 19, an embodiment of a Conversation Server 102 that implements the methods described above includes one or more processing units (CPU's) 1902, one or more network or other communications interfaces 1904, memory [[1006]] 1906, and one or more communication buses 1908 for interconnecting these components. The Conversation Server 102 may optionally include a user interface comprising a display device and a keyboard. Memory 1906, or a portion thereof, comprises a computer readable storage medium and may include high speed random access memory and may also include non-volatile memory, such as one or more magnetic or optical storage disks. Memory 1906 may include mass storage that is remotely located from CPU's 1902. The Memory 1906 may store:

[00152] a Delete/Remove Module [[1095]] 1952 for marking message or conversation for deletion and subsequently removing them as described earlier'